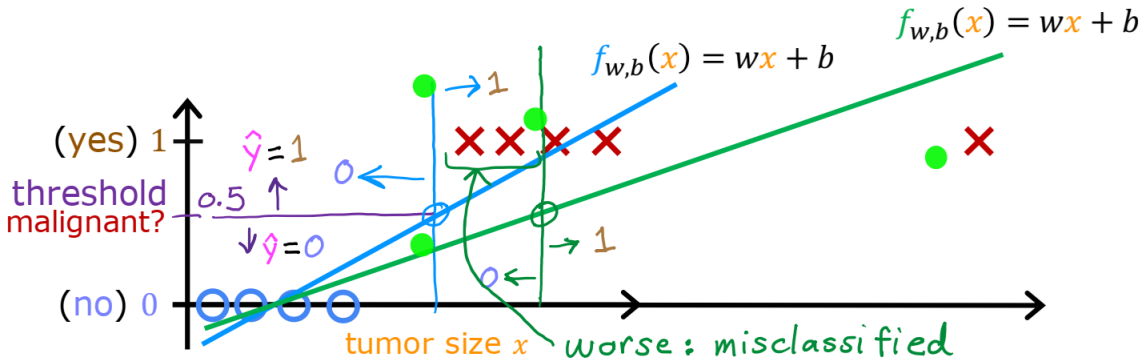
DeepLearning.AI Stanford ONLINE

SUPERVISED MACHINE LEARNING: REGRESSION AND CLASSIFICATION

MACHINE LEARNING SPECIALIZATION

LOGISTIC REGRESSION

- Logistic Regression solves classification problem by predicting output category from a group of classes. Categorization from only two output classes (Y/N, T/F) is Binary Classification.
- Decision Boundary separates class points, can be linear or non-linear lines
- Linear Regression function with threshold in straight line decision boundary misclassify new value as the best fit line shift to the left or right for new data.
- Logistic Function classifies regression model output or fit into classes based on threshold

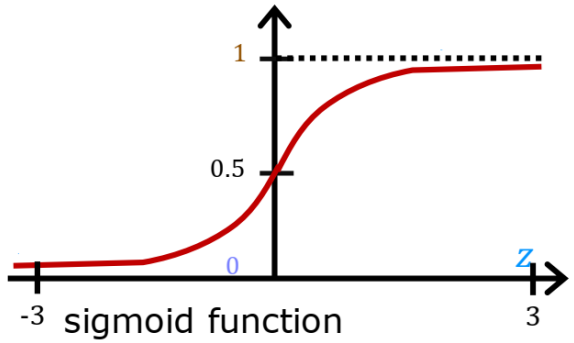


if $f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$

if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$

DeepLearning.AI Stanford ONLINE

LOGISTIC REGRESSION

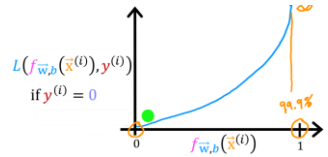


- Sigmoid Function output between 0 and 1, threshold can be 0.5
- Interpretation: $f_{wb}(x) = P(y = 1|x, w, b)$
- Linear Decision Boundary: $f_{(w,b)}(X) = w_1x_1 + w_2x_2 + b$
- Non-Linear Decision Boundary:

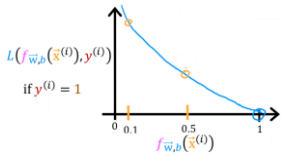
$$g(z) = \frac{1}{1+e^{-z}} \quad 0 < g(z) < 1$$

$$f_{\vec{w},b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1 + e^{-(\vec{w}\cdot\vec{x}+b)}}$$

$$f_{(w,b)}(X) = w_1{x_1}^2 + w_2{x_2}^3 + b$$
$$f_{(w,b)}(X) = w_1x_1 + w_2x_2 + w_3{x_1}^2 + w_4{x_2}^2 + w_5x_1x_2 + b$$



- Logistic Regression hypothesis feds Linear hypothesis into Sigmoid Function as output [0,1]
- Cost Function: Squared error cost function with Sigmoid function input creates wiggle non-convex curve with lots of local minima. Log loss function is used to predict true Y label with –log(f_wb (X)) function and false Y label with –log(1–f_wb (X)) function where loss is close to zero as Y=1 and Y=0.



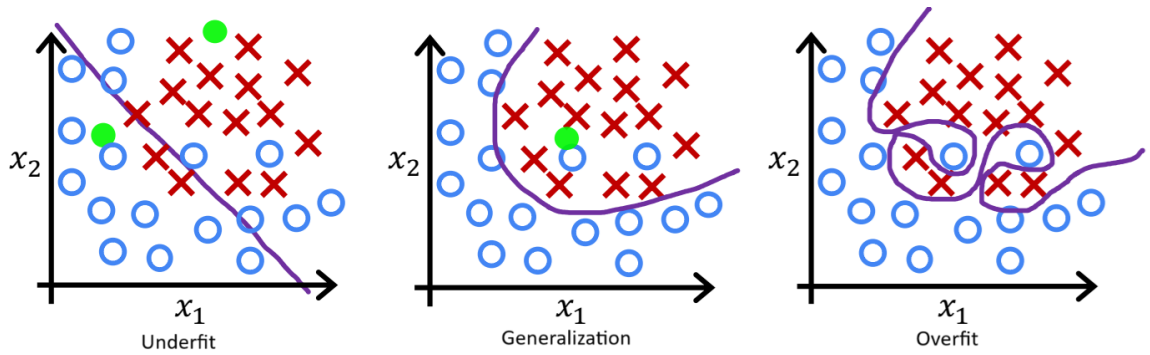
$$J_{wb} = \frac{1}{m}\sum L\left(f_{wb}\left(x^i\right), y^i\right) = -\frac{1}{m}\sum\left(y^i \log\left(f_{wb}\left(x^i\right)\right) + \left(1 - y^i\right)\log\left(1 - f_{wb}\left(x^i\right)\right)\right)$$

OVERFIT & UNDERFIT

- **Gradient Descent:** Logistic Regression weight gradients similar to Linear Regression

$$w = w - \alpha \frac{\partial}{\partial w} J_{(w,b)} = w - \frac{\alpha}{m} \Sigma (f^i_{(w,b)} - y^i) x^i \qquad b = b - \alpha \frac{\partial}{\partial b} J_{(w,b)} = b - \frac{\alpha}{m} \Sigma (f^i_{(w,b)} - y^i)$$

- **Underfit:** Model does not fit well in training data, low training accuracy, high loss and high bias, poor decision boundary and simple classification
- **Generalization:** Good prediction with high training and test accuracy
- **Overfit:** Model fits training set very well but fails in test set, high variance (polynomial function), extreme complex decision boundary, over classification. High variable in prediction for small changes in training set. **Reason:** Too many features and low data



REGULARIZATION

- Address Overfit: Collect more training data, Select features only that impact output (chances of loosing useful feature), Regularization (Large number of features)
- Regularization: Reduce the impact or effect of some features in hypothesis by applying smaller weight values which is less likely to overfit. Update cost function by adding weight parameters to minimize weight values. Regularization needed when number of feature is large (100~1000).
- Regularized Cost Function: $J_{(w,b)} = \frac{\sum(f_{wb}(x)-y)^2}{2m} + \frac{\lambda}{2m}\sum w_j^2$ (Regularization Term)
- λ is the regularization parameter which keep balance between fitting data and reducing overfit
- λ is too small then overfit and λ is too large then underfit
- Logistic and Linear Regression cost function derivative for weights with regularization

$$\frac{\partial}{\partial w_j} J_{(w,b)} = \frac{1}{m}\sum\left(f^i_{(w,b)} - y^i\right)x^i + \frac{\lambda}{m}w_j$$

- Weight w_j shrink with regularization by $(1 - \frac{\alpha\lambda}{m})$ in each iteration.